Modernisierung von BI-Architekturen für Big Data

Hinter »Big Data« verstecken sich eine Reihe altgedienter methodischer und technischer Ansätze, die auch in BI-/DWH-Architekturen genutzt werden. Dieser Artikel beleuchtet gängige Fälle, die eine Modernisierung der BI-Landschaft erfordern, und zeigt Technologien und Architekturen zu deren Umsetzung auf.

In diesem Beitrag erfahren Sie:
- welche häufigen Anwendungsfälle sich mit Big Data bewältigen lassen,
- wie sich die unter »Big Data« zusammengefassten Technologien gruppieren lassen sowie
- welche Best-Practice-Big-Data-Architekturen für bestehende BI-Landschaften es gibt.

Christopher Thomsen

Definition

Wer nach einer Definition von »Big Data« sucht, wird oft von den »drei Vs« lesen. Diese umfassen die Kernaspekte von Big Data und die Dimensionen, die Big-Data-Technologien zu bewältigen haben: *Volume* (Volumen), *Velocity* (Schnelligkeit) und *Variety* (Vielfalt) [1]. Diese Begriffsbestimmung ist jedoch ein eher akademischer Ansatz, um Big Data von Small Data abzugrenzen. Bei dem, was auf dem Produktmarkt vielfach unter Big Data verstanden wird, sind diese Kriterien oft nicht erfüllt. Warum ist das so? Big Data ist zwar ein IT-Trend, wird aber nicht primär von der IT getrieben, sondern von den Fachbereichen, die mit Big Data neue Möglichkeiten und Geschäftsmodelle verknüpfen. Die Umsetzung stellt technisch nicht immer ein Novum dar, durch ihre plötzliche Popularität werden sie vom Markt dennoch häufig als Produkt- bzw. Technologieneuheit wahrgenommen.

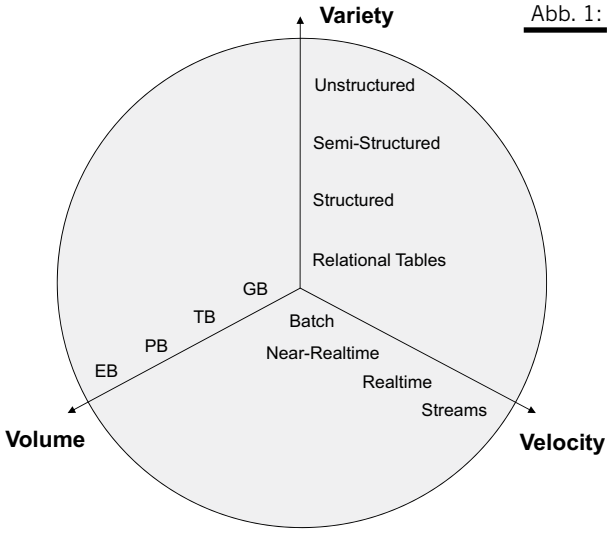

Abb. 1: *Dimensionen und Ausprägungen der drei Vs.*

Use Cases

Big-Data-Projekte beginnen also üblicherweise mit dem Use Case und nicht mit der Technologie. Um etwas Struktur in die Vielzahl der Big-Data-Anwendungsfälle zu bringen, lassen sich diese in verschiedene Strömungen einteilen, die für unterschiedliche Branchen von unterschiedlicher Relevanz sind.

Explorative Analysen

Explorative Analysen bieten die Möglichkeit, an einer Stelle im Unternehmen auf die Gesamtheit der zur Verfügung stehenden internen und externen Daten zuzugreifen. Damit bringen sie dem Unternehmen neue Erkenntnisse über Markt, Kunden, Produktion, Lieferketten und vieles mehr.

Voraussetzung für explorative Analysen ist, einen sogenannten Data Lake zu schaffen – ein einzelnes skalierbares Datenhaltungssystem, in dem große Mengen polystrukturierter Daten gehalten und verarbeitet

werden können. Auf den Data Lake können Analysten über eine *Self-Service-BI-Strategie* selbst zugreifen. Natürlich setzen diese explorativen Analysen das notwendige Know-how zur statistischen Modellbildung und deren Tests voraus. In diesem Zusammenhang wurde die Rolle des Data Scientist neu geschaffen, einer Person, die statistische Fähigkeiten und das Wissen, wie man Big-Data-Analysewerkzeuge nutzt, auf sich vereint.

Customer Intelligence

Big Data erfindet *Cross- & Up-Selling, Churn Prevention, Service-Value-Berechnung, Lead-Management* und die vielen anderen Aspekte der Customer Intelligence nicht neu, bringt aber neue Möglichkeiten mit sich, um die Güte der dahinter liegenden Modelle durch leistungsstärkere Modellbildungsverfahren sowie die Nutzung neuer semi- und unstrukturierter Datenquellen wie Social Media, Mailverkehr, Sensor-

Abb. 2: *Customer-Intelligence-Disziplinen*

daten oder Serverlogs zu verbessern. Auf ein Sampling der Daten kann damit verzichtet werden.

Intelligente Sicherheit

Während es bei den zwei vorhergehenden Beispielen primär um die Verarbeitungskapazitäten und Datenformate ging, zählt für intelligente Sicherheit vor allem die Fähigkeit, *Datenströme in Echtzeit* zu verarbeiten und *trainierte Muster* in diesen zu erkennen. Die Anwendungsfälle reichen von der Erkennung von Handydiebstählen oder kostenintensiven Umleitungen von Anrufen ins Ausland (z. B. aufgrund einer gehackten Telefonanlage) bis zur Feststellung von DDoS-Angriffen, von Account-Missbräuchen oder von der Nutzung von Bots auf der eigenen Plattform.

Operations Analysis

Eng verwoben mit weiteren Hype-Begriffen wie *Industrie 4.0* und *Internet der Dinge* stehen bei Operations Analysis meist Sensor- und Logdaten im Mittelpunkt der Analysen. Zur automatisierten Optimierung und Steuerung von Produktionsprozessen werden die gesammelten Daten mit erkennbaren Ereignissen verknüpft – z. B. mit dem Ausfall einer Maschine – und Mustererkennungsmodelle trainiert. Diese Modelle werden in Echtzeit auf neue Daten angewandt, um z. B. den Energieverbrauch zu senken, Störungen frühzeitig zu erkennen oder auch nur die Nutzung zu messen und dem Kunden nutzungs- bzw. verbrauchsorientierte Rechnungen zu stellen.

DWH-Modernisierung

Als letzte Big-Data-Strömung sei die DWH-Modernisierung genannt. Als Anwendungsfall birgt diese zwar keine neuen Geschäftsmodelle in sich, für viele Unternehmen ist die Modernisierung jedoch wichtig, um die Erstellungsgeschwindigkeit, den zeitlichen Betrachtungsumfang,

die Feingranularität, die Flexibilität und die semantische Breite von Reports zu erhöhen. Das erklärt auch den großen Erfolg der SAP HANA.

Funktionale Domänen

Im Gegensatz zu den meisten BI- und ETL-Werkzeugen und zu relationalen Datenbanken, die als Allrounder auftreten und die Bearbeitung z. B. einer Transformationsstrecke dezidiert in einem einzigen Werkzeug ermöglichen, sind die meisten Hilfsmittel des Big-Data-Ökosystems Spezialistenwerkzeuge. Diese sind für spezifische Zwecke konzipiert und können erst in Kombination mit anderen Tools eine komplette Wertschöpfungskette abbilden. Sie alle lassen sich gemäß ihres Einsatzzwecks in Domänen unterteilen, die ihr Anwendungsgebiet beschreiben.

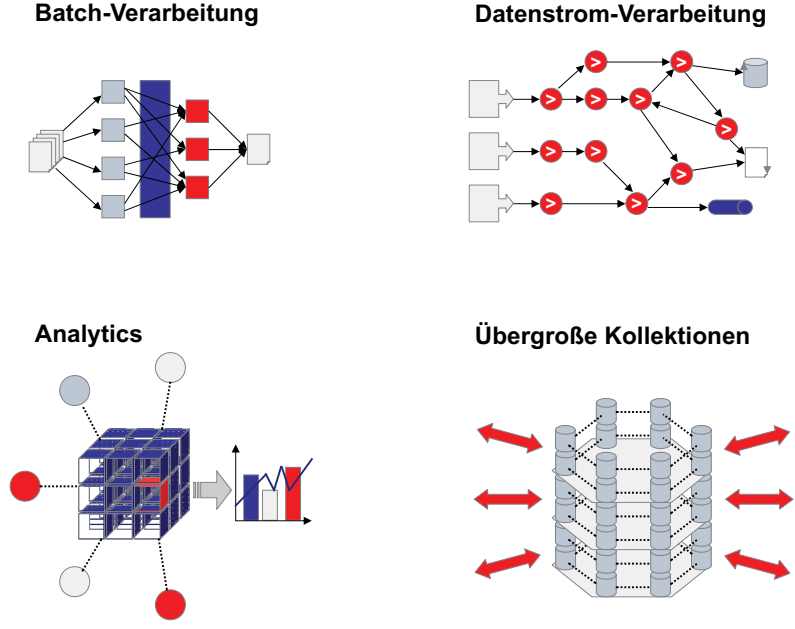

Abb. 3: *Funktionale Domänen von Big-Data-Technologien*

Batch Processing

Die meisten ETL-Werkzeuge führen ihre Datenbewirtschaftungsstrecken als Batch-Operationen aus. Auch Hadoop, das wohl bekannteste Big-Data-Skalierungsframework, das hier später noch detaillierter betrachtet wird, wurde ursprünglich ausschließlich als Batch-System konzipiert. Alle Batch-Processing-Werkzeuge des Big-Data-Ökosystems haben eines gemeinsam: Sie sind »scale out«-fähig. Das bedeutet, dass sie sich horizontal skalieren lassen, um die Rechenlast auf mehrere Rechenknoten zu verteilen. Optimalerweise ermöglicht dies die lineare Skalierbarkeit der Rechenleistung.

Werkzeuge dieser Domäne können meist datenformatunabhängig arbeiten und sind so in der Lage, auch hierarchisch angeordnete und textuelle Daten zu verarbeiten. Das bekannteste Framework zur Batch-Verarbeitung ist das bereits genannte Hadoop-Framework, auf dem eine Vielzahl von Open-Source-Lösungen und kommerzielle Produkte aufsetzen. Hadoop wird zusammen mit anderen Produkten in kommerziellen Distributionen angeboten.

Stream Processing

Im Gegensatz zum Batch Processing kommen Stream-Processing-Werkzeuge dort zum Einsatz, wo Anwender Realtime- oder Near-Realtime-Anforderungen haben und direkt auf eingehende Daten reagieren wollen, wie es bei Sicherheitssystemen, Maschinenüberwachung oder Realtime Dashboards der Fall ist. Diese Systeme verarbeiten Datensätze – im Streaming-Kontext »Tupel« genannt – isoliert voneinander durch eine Reihe von Knoten. Die Verarbeitung erfolgt meist mit In-Memory-Systemen und durch Micro Batches. Die bekanntesten Open-Source-Frameworks zur Datenstromverarbeitung sind Apache Storm und Apache Spark Streaming. Auf kommerzieller Seite sind IBM InfoSphere Streams, Oracle Event Processing, SAP Event Stream Processing, Informatica Event Processing und TIBCO StreamBase die präsenten Vertreter.

Big Collections

Keine der beiden zuvor genannten Domänen erfüllt die Aufgabe einer konventionellen Datenbank, Ad-hoc-Abfragen in angemessener Zeit zu bedienen. Schnell stellt man fest, dass die Antwortzeiten deutlich hinter dem liegen, was man von relationalen Datenbanken gewohnt ist. Doch der Schein trügt: Der Datendurchsatz der Batch-Jobs auf diesen skalierbaren Systemen ist zwar tatsächlich höher, doch gleichzeitig ist der Ausführungsplan auch deutlich ineffizienter gestaltet als bei einer konventionellen Datenbank. Das liegt daran, dass diese Systeme stets alle Daten vollständig einlesen und verarbeiten und dabei keine Indizes verwenden. Erschwerend kommt die oft zeitintensive Vorbereitung für die verteilten Jobs hinzu, die unabhängig von der Datenmenge besteht.

Diese Umstände machen diese Art der Verarbeitung für Ad-hoc-Abfragen oder kleine bis mittelgroße Datenmengen ungeeignet. Zwar gibt es z. B. in Hadoop mit Apache Hive ein Werkzeug, das tabellarisch organisierte Daten mit einer SQL-ähnlichen Sprache anbietet, doch sollte dem Anwender bewusst sein, dass dieses Werkzeug im Hintergrund völlig anders arbeitet als eine SQL-Datenbank. Sollen eine Vielzahl an Anfragen in kürzester Zeit durchgeführt werden, die nur einzelne Datensätze des gesamten Bestandes zurückliefern und die hinsichtlich Datenmenge, Spaltenanzahl oder Tabellenlänge die Kapazitäten einer einzelnen relationalen Datenbank sprengen, so spricht man von sogenannten Big Collections. Je nach Art der Daten existieren für diese Szenarien spezielle NoSQL-Datenbanken, die ihre Daten Key-Value-, spalten-, dokument- oder graphenorientiert ablegen, die horizontal skalierbar sind und mit steigender Datenmenge durch das Hinzufügen weiterer Datenbankknoten mitwachsen. Erwähnenswerte Vertreter unter den zahlreichen NoSQL-Datenbanken sind Redis für In-Memory Key Value Stores, HBase und Cassandra für die Column Family Stores, MongoDB für dokumentenorientierte Datenbanken und Neo4j für Graphendatenbanken.

Advanced Analytics

Dank Big Data erleben Statistikapplikationen wie MatLab, Weka und R im Moment als Advanced-Analytics-Werkzeuge einen zweiten Frühling. Durch die Integration in skalierbare Batch-Frameworks wie Hadoop oder In-Memory-Plattformen wie SAP HANA ist die Berechnung wesentlich umfangreicherer Modelle möglich. Advanced Analytics umfasst hierbei Verfahren, die über das BI-Standardreporting hinausgehen und vor allem prädiktive Auswertungen ermöglichen.

Analytische Szenarien

Branchenübergreifend haben sich im Rahmen des Big-Data-Hypes einige analytische Lösungsszenarien herauskristallisiert, die datengetriebene Unternehmen dabei unterstützen, zukünftige Ereignisse vorherzusagen und aus diesen Vorhersagen die sinnvollen nächsten Schritte abzuleiten.

Churn Prevention

Die Segmentierung von Kunden in loyale Ansprechpartner einerseits und in solche, die ihren Vertrag mit der Firma wahrscheinlich kündigen werden andererseits, ist keine neue Erfindung. Das Hinzuziehen weiterer Datenquellen wie Social-Media-Kurznachrichten, das Verhalten bei der Nutzung von Produkten und Clickstreams sowie die Möglichkeit, deutlich umfangreichere Regressions- und Clustering-Analysen durchzuführen, ermöglichen jedoch die exaktere und feingranularere Bestimmung des Churn-Potenzials einzelner Kunden. Auf Grundlage der Analysen können Vertriebsmitarbeiter Kunden von durch Abwanderung gefährdeten Segmenten beispielsweise gezielt mit Lockangeboten ansprechen, um sie als Kunden zu halten.

Customer Lifetime Value

In die gleiche Sparte fällt auch der Customer Lifetime Value. Hierbei geht es um den Umsatz, der mit dem Kunden über seine Lebenszeit voraussichtlich generiert werden wird. Diese Informationen sind vor allem bei Neukunden relevant, zu denen noch keine eigenen Umsatzdaten vorliegen. Für diese Kundengruppe lassen sich anhand aller Daten, die über diese Person zugänglich sind, Korrelationen zu Bestandskunden mit vorhandenen Umsatzdaten errechnen. Auf diese Weise können Vorhersagen über den weiteren Umsatz getroffen werden.

Customer Segmentation

Auch das Marketing kann alle vorhandenen Datenquellen für die Segmentierung von Kunden und Leads nutzen und auf dieser Basis zielgerichtete Kampagnen aufsetzen. Die Customer Segmentation ist dank Textanalysewerkzeugen auch auf unstrukturierten Textdaten möglich. So können z. B. anhand der Wortwahl in Texten, anhand von Inhalten gelesener Dokumente oder anhand von Keywords aus E-Mails und sonstiger Korrespondenz Dimensionen zur Segmentierung abgeleitet werden.

Next Best Action

Auf der Kundensegmentierung aufbauend besteht ein weiteres Analyseziel für den Vertrieb eines Unternehmens darin, das beste Angebot zur richtigen Zeit zu machen – also die »Next Best Action« einzuleiten. Dazu werden neben der Kundensegmentierung die Daten zu früheren Angeboten mit unterschiedlichem Vorlauf und die dazugehörenden Erfolgsraten benötigt. Diese Informationen können für ein gezieltes Up- und Cross-Selling genutzt werden, bei dem der Verkäufer einem ausgewählten Kunden nicht nur Einzelprodukte, sondern den für ihn persönlich relevanten Produktmix empfiehlt.

Product Propensity

Auch nach dem Kauf kann der Analyst wichtige Daten ermitteln: Setzt man das Verhalten des Kunden in sozialen Netzwerken oder beim allgemeinen Surfen im Internet in Relation zu seinem Kaufverhalten, hilft dies zu verstehen, was einzelne Kundengruppen beim Kauf eines Produkts beeinflusst. Dieser Use Case nennt sich »Product Propensity« und hilft dabei vorherzusagen, was ein Kunde kaufen wird, noch bevor er selbst es weiß.

Sentiment Analysis

Ein meist in Kombination mit den bereits genannten Szenarien genutztes Analyseverfahren ist die Sentiment Analysis oder zu Deutsch: Tonalitätsanalyse. Diese Analyse ermöglicht es, die Stimmung des Autors aus einem Textauszug abzuleiten. Während die Granularität der Analyse meistens nur in drei Ausprägungen – negativ, neutral und positiv – ausgegeben wird, gibt es hinsichtlich des Tonalitätsgegenstands verschiedene Abstufungen. So kann die Tonalität einer Aussage eine andere sein als die Tonalität eines Terms im Dokument. Beides kann im jeweiligen Anwendungsfall relevant sein.

Die Gesamttonalität spielt zum Beispiel bei der inhaltsunabhängigen Ermittlung der Stimmungen von Kundenkorrespondenz eine wichtige Rolle. Auf diese Weise lassen sich unzufriedene Kunden ermitteln oder Probleme in der eigenen Supportstruktur identifizieren. Das termbezogene Sentiment zeigt hingegen, wie Marken oder Produkte im Social-Media-Umfeld bewertet werden.

Predictive Maintenance

Predictive Maintenance ist ein Big-Data-Anwendungsfall, der nicht aus dem Vertragskundengeschäft stammt, sondern vor allem in der Industrie zu beobachten ist. Über die Korrelation von historischen Sensordaten mit Ausfällen und Störungen trainiert ein System Modelle zur

Mustererkennung. Ziel ist es, auf einem Datenstrom mit der gleichen Sensoranordnung Muster zu erkennen, nach denen in signifikanter Häufung eine Störung folgt. Mithilfe dieser Muster können Techniker ein Gerät warten oder ein Bauteil ersetzen, bevor es zu einem Ausfall kommt.

Quality Assurance

Ähnlich verhält es sich mit Qualitätsproblemen, die zu Unzufriedenheit beim Kunden führen können. Durch Modelle zur frühzeitigen Erkennung systematischer Fehler anhand von Sensordaten und Kundenfeedback und der Bewertung ihrer Relevanz können wichtige Informationen für den Quality-Assurance-Prozess gewonnen werden.

Hadoop

Hadoop, dessen verteiltes Dateisystem HDFS und MapReduce sind meist die ersten Technologien, mit denen Interessierte beim Thema Big Data in Berührung kommen. Die Ernüchterung folgt meist auf dem Fuß: Für die Analyse von Social-Media-Inhalten und Ähnlichem stellen die Standard-Tools von Hadoop keine Funktionen bereit. Auch die Antwortzeit bei der wahlfreien Suche oder gar bei verknüpften, analytischen Abfragen über mehrere Tabellen ist weit entfernt von einem Echtzeitansatz und unterliegt den Ergebnissen einer leistungsstarken relationalen Datenbank.

Doch eines ist klar: Hadoop verfehlt diese Erwartungen, weil es nicht primär für solche Zwecke konzipiert wurde. Hadoop kann in zweierlei Weise betrachtet werden: zum einen als eine Technologieplattform für die verteilte Batch-Verarbeitung, auf der andere Technologien aufsetzen; zum anderen als eine Staging-Umgebung in einer ETL-Strecke zur Aggregation von Rohdaten, die in einem nachgelagerten DWH zwecks Darstellung und Auswertung abgelegt werden.

Hadoop ist dabei keine einzelne Applikation, sondern eine Plattform, die üblicherweise mit einem Set grundlegender Werkzeuge für skalierbare Datenverwaltung und -verarbeitung ausgeliefert wird. Der

Kern von Hadoop, der von der Apache Foundation betreut und entwickelt wird, kann separat installiert und manuell um einzelne Werkzeuge ergänzt werden. Für den produktiven Einsatz werden jedoch meist vorgefertigte, teils kommerzielle Distributionen mit teilweise proprietären Anteilen eingesetzt [2].

Distributionen

Am Markt haben sich eine Reihe von Softwareherstellern auf Hadoop spezialisiert. Die meisten von ihnen setzen auf eine der fünf etablierten Hadoop-Distributionen: Cloudera, MapR, HortonWorks, IBM InfoSphere BigInsights und Pivotal HD. Am deutschen Markt besitzt HortonWorks einen vergleichsweise hohen Marktanteil, was an der Kooperation mit deutschen Softwareherstellern wie SAP liegen könnte. Unabhängig von der Distribution werden einige Werkzeuge des Kernökosystems von Hadoop stets mit ausgeliefert. Diese sollen im Folgenden kurz beleuchtet werden.

Hadoop Distributed File System (HDFS)

Die Bewältigung immer größerer Datenmengen stellt eine zentrale neue Herausforderung dar. 2005 gab es weltweit ca. 130 Exabyte an digitalen Daten, 2010 hatte sich diese Zahl bereits verzehnfacht [3]. Eine ähnliche Entwicklung schlug in der gleichen Zeit auch die Rechenleistung ein. Doch wenn sich die Rechenleistung und der Zuwachs an Daten die Waage halten, warum sollten Unternehmen dann neue Wege für die Handhabung der Daten suchen?

Den Unterschied macht nicht die reine Datenmenge, sondern deren Nutzung. Denn was nützen große Datenmengen, wenn kein Mehrwert daraus generiert werden kann? Was Unternehmen interessiert, sind Konsumententrends, zusätzliche Verbraucherinformationen und Meinungsbilder. Damit können sie Trends frühzeitig erkennen, Imageprobleme angehen und direkt vom Kunden erfahren, was sie besser machen können, um langfristig wettbewerbsfähig zu bleiben

und steigende Umsätze und Gewinne zu erzielen. Doch diese Informationen sind selten direkt in den Daten zu finden, sondern liegen in den Beziehungen zwischen ihnen. Um diese zu ermitteln, muss der gesamte Datenbestand vorgehalten und bei Bedarf in kurzer Zeit verarbeitet werden können.

Das HDFS ist derzeit der Spitzenreiter unter den verteilten Dateisystemen. Das Vorgehen des HDFS besteht in der Verteilung von Datenblöcken über mehrere Systeme. Hierbei stellt die Software die Redundanz und die Allokation der Daten sicher. Das System macht es möglich, auf einer einzelnen virtuellen Partition Daten jeglicher Formate zu verwalten. So können große Datenmengen, bei denen einfache RAID- und Computersysteme an ihre Grenzen stoßen, verteilt auf zahlreichen Systemen verwaltet und verarbeitet werden.

YARN und MapReduce

In den Versionen 0.x und 1.x war das HDFS zur verteilten Verarbeitung von Daten direkt mit dem MapReduce-Algorithmus von Google verzahnt. Dieser ermöglichte es, eine Vielzahl an Rechenoperationen nahezu beliebig ausgedehnt in einem Cluster zu verteilen. So konnten die Rechenzeiten von rechenintensiven Batch-Operationen drastisch reduziert werden. Der hohe, volumenunabhängige Overhead, die starre Trennung in Map- und Reduce-Phase sowie die fehlende Unterstützung von In-Memory-Operationen zwang die Entwicklung jedoch dahin, den Mechanismus zur Ressourcenallokation im Cluster zu öffnen. Auf diese Weise wurde die Entwicklung weiterer Algorithmen auf Basis von Hadoop ermöglicht.

Seit Hadoop 2.0 übernimmt YARN diese Aufgabe. Dazu verwaltet es lediglich die Ressourcen des Clusters, die laufenden Jobs sowie den Zugriff auf das HDFS, hat jedoch keinen expliziten Algorithmus zur Ausführung der Jobs hinterlegt. Dies ermöglicht die Nutzung verschiedener Algorithmen für die Ausführung oder sogar die Entwicklung eigener Verteilungs- und Rechenverfahren auf Basis von Hadoop. HortonWorks setzt auf Tez und Slider als Alternativen zu MapReduce

mit deutlich kleinerem Overhead und hat Ersteres auch bereits in die Werkzeuge Pig und Hive integriert. Cloudera setzt mit Impala auf ein indiziertes Verfahren für SQL-ähnliche Abfragen. IBM entwickelte schon früh mit Adaptive MapReduce eine eigene Alternative zu Googles MapReduce, die eine deutlich bessere Skalierbarkeit, eine geringere Latenz und eine Priorisierung von Aufgaben ermöglicht.

Pig

Die mit Hadoop ausgelieferte Skriptsprache zum einfachen Schreiben von verteilten Transformationsketten ist Pig Latin. Diese wird vom gleichnamigen Pig Interpreter in MapReduce-Jobs anderer Algorithmen (z. B. Tez unter HortonWorks) übersetzt und ausgeführt.

Listing 1: Wordcount-Aggregation in Pig.

```
-- Wordcount implementiert in Pig
input = LOAD ‚/hdfs/pfad/zu/dateien‘ AS (line:chararray);
words = FOREACH input GENERATE FLATTEN(TOKENIZE(line)) AS word;
filtered_words = FILTER words BY word MATCHES ‚\\w+‘;
word_groups = GROUP filtered_words BY word;
word_count = FOREACH word_groups GENERATE COUNT(filtered_words) AS
count, group AS word;
ordered_word_count = ORDER word_count BY count DESC;
DUMP ordered_word_count;
```

Hive

Zum Standard nahezu jeder Hadoop-Installation gehört mittlerweile Hive, eine tabellarische, jedoch nicht relationale Datenbank mit einer SQL-ähnlichen Abfragesprache (HiveQL). Diese kompiliert MapReduce-Jobs zur Ausführungszeit. Hive legt für jede angelegte Tabelle einen eigenen Ordner im HDFS an. Gleiches gilt für Hive-Partitionen, die für jede Attributausprägung einen eigenen Subordner erhalten. Hive-Partitionen sind Tabellenattribute, welche die Metainformation »Partition« erhalten. Dies kommt einer Indexierung in einer konven-

tionellen Datenbank nahe. Hive unterstützt verschiedene Serializer für die Tabellendaten und ist durch eigene Serializer erweiterbar.

Die einfachste, aber auch zugleich ineffizienteste Art der Serialisierung, die Hive unterstützt, ist zeichenseparierter Klartext. Vorteilhaft bei dieser Anwendung ist, dass Daten direkt im Ordner von anderen Anwendungen manipuliert oder geschrieben werden können. HiveQL ist allerdings hinsichtlich seiner Transformationsfähigkeiten limitiert und deckt den ANSI-SQL-Standard mit seinen Funktionen nicht ab.

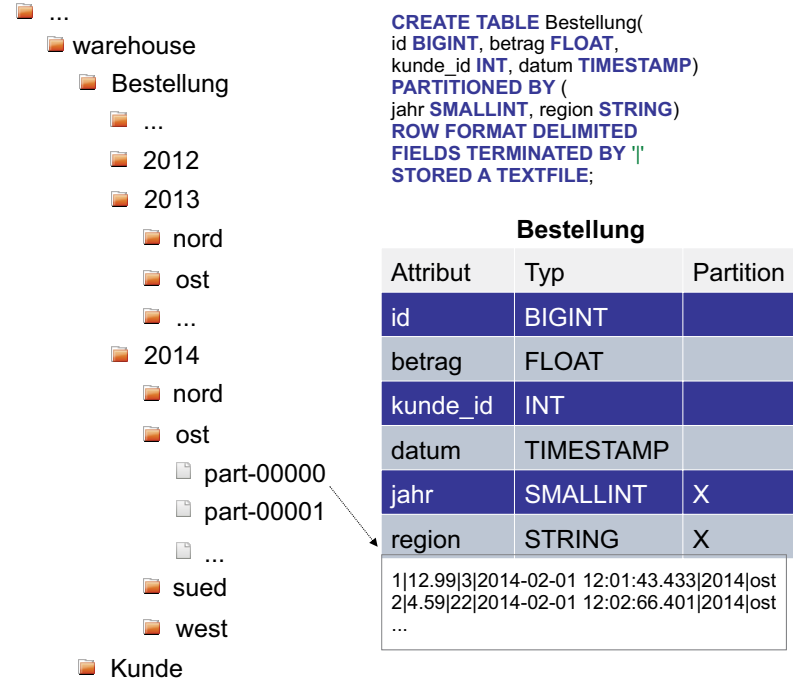

Abb. 4: *Serialisierung einer partitionierten Hive-Tabelle auf dem Dateisystem*

HBase

Hive ist nicht wie eine gewöhnliche Datenbank ausgelegt und daher für Ad-hoc-Abfragen nicht geeignet. Hier bieten NoSQL-Datenbanken die passende Technologie, um wahlfreien Zugriff auf Big Collections zu ermöglichen. Hadoop liefert HBase als Column Family Store in der Hadoop-Distribution mit aus. Column Family Stores sind ähnlich aufgebaut wie eine Key Value Map, mit dem Unterschied, dass einzelne Spalten des Values zu Column Families zusammengefasst werden können.

Hive ist in der Lage, seine tabellarisch organisierten Tabellen auf HBase Collections zu projizieren und damit eine SQL-ähnliche Schnittstelle für HBase bereitzustellen. Dies ermöglicht im Gegensatz zur direkten Serialisierung auf dem HDFS u. a. die Möglichkeiten inkrementeller Datenupdates oder des selektiven Löschens einzelner Datensätze und führt zu einer deutlich geringeren Latenz bei Ad-hoc-Abfragen. Wie die meisten NoSQL-Datenbanken kennt HBase jedoch keine Fremdschlüssel. Nur der Key wird indiziert, sodass sich Joins zwischen mehreren Collections vergleichsweise aufwendig gestalten.

Sqoop

Neben Hive als Schnittstelle mit SQL-ähnlicher Abfragesprache ist es auch über Sqoop möglich, Daten zwischen Hive-Tabellen oder tabellarisch organisierten Daten auf dem HDFS und einer externen relationalen Datenbank über JDBC bidirektional zu synchronisieren. Sqoop wird konfigurativ über das Command Line Interface (CLI) gesteuert und enthält keine eigene Skriptsprache.

Flume

Um Datenströme – seien es aus Streaming-Anwendungen, Logfiles, an denen permanent Zeilen angehangen werden, oder von operativen Anwendungen per TCP-Socket übertragene Daten – zuverlässig aus

verschiedenen Quellen in das HDFS zu transportieren, bietet Flume eine Agentenarchitektur, die über die Definition von Datenquellen, Kanälen und Datensenken die Schaffung von Ingress-Topologien ermöglicht. Sie übernimmt dabei den sicheren Transport, gewährt Ausfallsicherheit und erkennt Datenänderungen, zum Beispiel bei Logfiles.

Oozie

Zur Orchestration dieser und weiterer distributionsabhängiger Werkzeuge bietet Oozie eine limitierte Notationssprache, mit der sich automatisierte Workflows abbilden lassen. In Oozie-Workflows lassen sich alle genannten Hadoop-Werkzeuge ansprechen und konfigurativ ausführen. Zu jeder Aktion können vorbereitende und nachbereitende Aktivitäten wie das Löschen von Dateien oder das Erstellen von Ordnern auf dem HDFS ausgeführt werden. Oozie ist, was die Möglichkeiten der Gestaltung der Workflow-Topologie angeht, jedoch äußerst limitiert, sodass keine Schleifen, Events oder komplexe Operatoren möglich sind und sich die Übergabe von Hadoop Tools in den Prozess sehr unkomfortabel gestaltet.

Ergänzung klassischer BI-Architekturen

Mit Big Data rücken viele vernachlässigte BI-Themen wie Data Mining, Visualisierung sowie Autonomie und Flexibilität durch Self Service BI wieder in den Vordergrund. Big Data hat nicht zum Ziel, durch die hoch skalierbare Integration Datenverwaltung und Analyse polystrukturierter Daten-BI und -DWH in bestehenden BI-Strukturen zu ersetzen. Es geht vielmehr darum, diese zu ergänzen.

λ-Architektur

Streamingansätze und -Technologien wurden bereits besprochen. Sie decken jedoch keine analytischen Funktionen auf historischen Daten ab. Werden Analysen zu Datensätzen benötigt, die im Speicher nicht als Fenster eines Datenstroms abbildbar sind, im gleichen Zuge aber

eine hohe Aktualität in Realtime oder Near-Realtime besitzen, so ist die Integration der Streaming-Topologie in die DWH-/BI-Architektur notwendig. Die λ-Architektur bietet den am weitesten verbreiteten Ansatz zur zunächst separaten batch- und streamingorientierten Verarbeitung und dem Zusammenfügen der Views in gemeinsame Resultate.

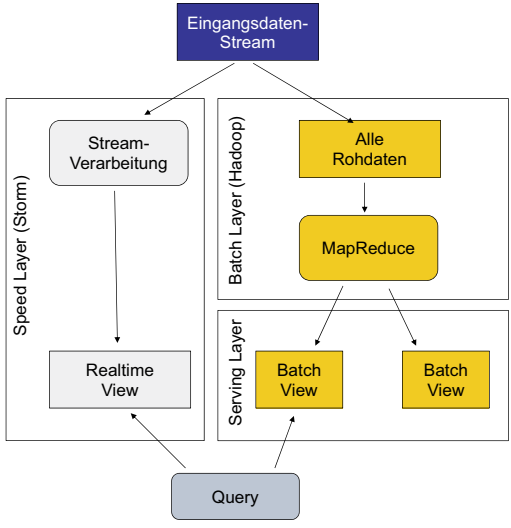

Abb. 5: *Schematische Abbildung der λ-Architektur*

Die λ-Architektur kann auf die klassische BI-Architektur projiziert werden, indem das Core Data Warehouse in Echtzeit mit vorgefilterten und voraggregierten Daten versorgt wird. Die Architektur überbrückt so die Lücke bis zum Abschluss des nächsten Batch-Laufs auf der vorgelagerten Staging-Umgebung. Gleichzeitig versorgt sie aus dem separaten Realtime-Prozess auch die Onlineanalyse und die Realtime Dashboards mit Daten und stößt als Reaktion auf ausgelöste Events wie erkannte Schlüsselwörter, Muster oder die Überschreitung von Grenzwerten weiterführende Events oder Unternehmensprozesse an.

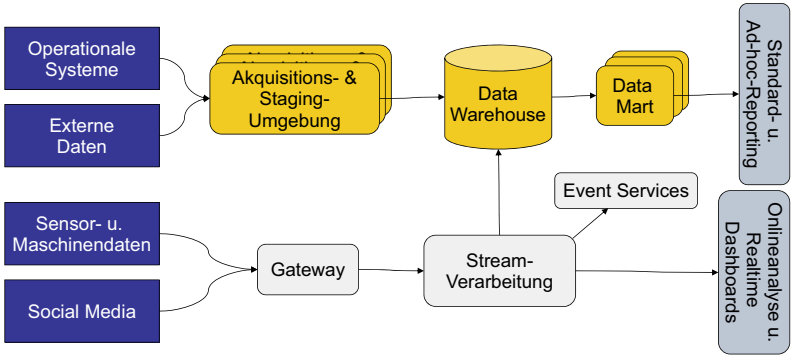

Abb. 6: *Streaming-Elemente in der klassischen BI-Architektur*

Analytics Lab

Analytische Plattformen können zusätzlich zum DWH eingesetzt werden, um mit nicht operativen Aktivitäten durch analytische Abfragen zu interferieren, massiv parallele Datenverarbeitung zu ermöglichen und integrierte analytische Funktionen zur Verfügung zu stellen.

Hadoop kann durch die Erweiterung mit kompatiblen analytischen Werkzeugen wie Mahout, R, Lucene oder Weka als Analytic Lab eingesetzt werden. Das Lab besteht parallel zur klassischen Reportingstruktur und kann in Kombination mit User Sandboxes ein Ansatz für eine Self-Service-BI-Architektur sein. Die Kehrseite dieses Ansatzes ist die

Einführung neuer Endnutzerwerkzeuge für die Arbeit auf dem Analytics Lab. Denn bislang ist kein Standard-BI-Werkzeug in der Lage, diese Aufgaben auf verteilten Plattformen wie Hadoop, Mesos oder Cassandra zu übernehmen.

Hadoop kann als analytische Plattform vorgelagert zum Data Warehouse auch für ein Preprocessing der Daten genutzt werden und die Voraggregation und die Transformation von nicht tabellarisch strukturierten Daten und großen Rohdatenaufkommen übernehmen. Preprocessing-Umgebung und Analytics Lab können technisch durchaus die gleiche Plattform sein, sofern für die benötigten Anwendungsfälle die gleiche Technologiebasis verwendet werden kann und soll. Dies entspräche dann als initiale Stage zur Verwaltung, Vorverarbeitung und Analyse von Rohdaten dem, was so oft als »Data Lake« bezeichnet wird.

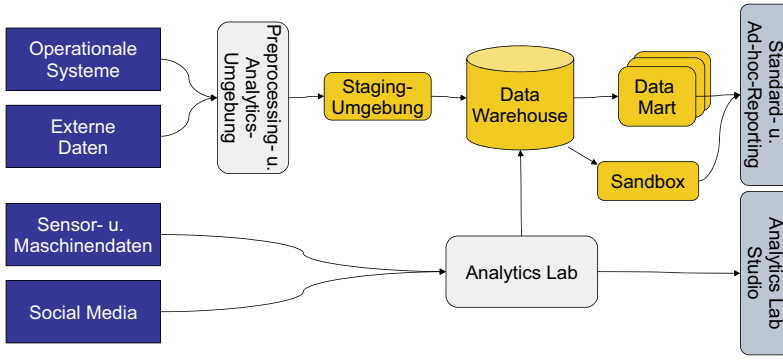

Abb. 7: *BI/DWH-Architektur mit Analytics Lab, Preprocessing Stage und Sandboxing.*

Onlinearchive

Wird für das Standardreporting keine Voraggregation gewünscht oder wird diese sehr feingranular durchgeführt, können die Storage-Kapazität und die damit verbundenen Kosten schnell zu einem Problem werden. Dieses Problem wird meist nach einer gewissen Zeit durch

die Archivierung und Löschung historischer DWH-Datensätze angegangen. Betrachtungen sehr langer historischer Zeiträume sind damit nicht mehr möglich.

Hadoop bietet neben seinen analytischen Funktionen auch einen sehr günstigen Massenspeicher in Form eines Onlinearchivs. Aus dem DWH zu archivierende Daten werden von Hadoop auf das HDFS oder auf ein anderes Werkzeug wie Hive oder Impala verschoben, das diese Daten tabellarisch organisiert. Viele Enterprise DWHs bieten bereits native Integrationsmöglichkeiten zu Apache Hive. So können sie bei SQL-Queries, die auf historische Datensätze zugreifen wollen, die nicht mehr im DWH liegen, die Anfrage an Hadoop weiterleiten und die Daten aus diesem Archiv beziehen. Diese Form des Late Binding führt bei historischen Abfragen verständlicherweise zu deutlich höheren Latenzzeiten, bietet jedoch die Möglichkeit für den Endnutzer, transparent Daten über beliebig lange Zeiträume kostengünstig und jederzeit zugreifbar vorzuhalten.

Die richtige Zeit für die BI-Modernisierung

Steigende fachliche und technische Anforderungen erfordern integrierte Analysetopologien durch modere Architekturansätze und den Einsatz skalierbarer Technologien. Durch ihre Vielfältigkeit und Komplexität rücken zukünftig integrierte Analyselandschaften sowie deren Flexibilität und Stabilität in den Vordergrund. Dezentralität sowie Integration mit operationalen Systemen durch Serviceintegration nehmen damit Einzug in die BI-Welt.

Auch wenn der Big-Data-Markt nach wie vor sehr volatil ist, kristallisieren sich doch industriefähige Ansätze zur Modernisierung von bestehenden BI-Architekturen heraus. Dies mag weniger disruptiv wirken, als man es vom derzeitigen Big-Data-Hype erwarten mag, doch eine stabile Überführung der BI-Architektur hin zu einer integrierten Analyselandschaft bringt langfristig neue Möglichkeiten zur Optimierung und Abbildung neuer Geschäftsmodelle und zu deren Wertschöpfung mit sich.

Literatur

[1] Kreuzer, R. T.; Land, K.-H.: *Digitaler Darwinismus: Der stille Angriff auf Ihr Geschäftsmodell und Ihre Marke. Springer Gabler Fachmedien, 2013*

[2] Wartala: *Hadoop – Zuverlässige, verteilte und skalierbare Big-Data-Anwendungen. Open Source Press, 2012*

[3] *www.statista.com (letzter Zugriff am 18.02.2015)*

Zusammenfassung

Unter »Big Data« zusammengefasste Technologien sind meist kein technisches Novum, sondern lediglich eine Zusammenführung unterschiedlicher Spezialistenwerkzeuge zur Bewältigung von Daten von großem Volumen, kurzlebigem Informationswert und vielfältiger Struktur. Diese Werkzeuge lassen sich anhand ihrer funktionalen Ausrichtung in vier Disziplinen untergliedern. Trotz dieser Vielfältigkeit ist Big Data keine Allzweckwaffe, sondern wird produktiv bislang meist nur in einigen wenigen Szenarien angewandt. Zur Bewältigung dieser Anwendungsszenarien unter Beibehaltung der etablierten BI-Systeme haben sich Best-Practice-Architekturen mit Big-Data-Bausteinen entwickelt.